BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/678,511	10/03/2000	Elliot Omiya	MSFT-0231/160306.1	5346

7590 07/30/2004

Peter M Ullman
Woodcock Washburn Kurtz
Mackiewicz & Norris LLP
One Liberty Place 46th Floor
Philadelphia, PA 19103

EXAMINER	
STEELMAN, MARY J	

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 07/30/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/678,511	OMIYA ET AL.	
	Examiner	Art Unit	
	Mary J. Steelman	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 April 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-50* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to Amendment and Arguments dated 26 April 2004. Claims 1, 2, 3, 5, and 50 have been amended. Claims 1-50 are pending.

Claim Rejections - 35 USC § 112

2. In view of the amendment to claim 50, the prior 35 USC 112 2nd paragraph rejection is hereby withdrawn.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. **Claims 1-4** are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent 6,237,135 to Timbol.

Per claim 1:

-a base object having:

(Timbol: Col. 4, lines 20-22, "...specify the name of the bean, the package it will be in, and the class (base object) it extends from." Col. 4, lines 25-26, "...choose a class to extend (from a base class)...", col. 9, lines 33-63, "The JAVA Bean...architecture or component model...provides standard design patterns...for components...the...model specifies a set of

design patterns of how component should be coded...For instance, if one has an 'account balance' JAVA bean, there would also exist an "account balance JAVA bean info class...(a base object)")

-internal logic executable on a computing device, said internal logic causing aid computing device to perform one or more actions, said one or more actions including the signifying of one or more events;

(Timbol: Col. 9, lines 64-67, "The present invention provides a wizard-based tool which automatically generates code to define property setters and getters, accessor methods, event listener and registration mechanisms and the like."

-a public object model which includes identifiable references to said one or more events;

(Timbol: Col. 9, lines 61-62, "...the bean info describes which properties are going to be exposed...(public)", col. 9, lines 64-67, "The present invention provides a wizard-based tool which automatically generates code to define property setters and getters, accessor methods, event listener and registration mechanisms and the like (public object model which includes identifiable references to said one or more events)", col. 11, lines 40-41, ...the class is declared as public...", col. 11, lines 49-53, "...properties usually have both read and write access methods (references to an event)...A getter method returns the current value of the property; a setter method sets the property to a new value.", col. 15, lines 12-14, A JAVA Bean can generate (or fire) events, sending an event object to a listening object, and can listen for events and respond to them when they occur.")

-a customization object having:

-data or logic representative of said public object model;

(Timbol: Col. 16, lines 59-67, "If the bean is registered with another component as a listener...the source component calls one of the methods in the listening component when that type of event occurs. For example, if a KeyPressed event occurs in the source component, the KeyPressed() method is called in the listening component. Therefore, if the user wants the component to respond in some way to such an event, the user simply writes the code (customization object) that responds (logic) within the body of the KeyPressed () method...")

-an event handler which receives the signified events from said base object, and which invokes at least one of a plurality of customized code sequences based on said data or logic, each of the at least one customized code sequences being separately retrievable from a source separate from said customization object.

(Timbol: Col. 17, lines 2-9, "...user might want to create a custom event set to describe other events that can occur in the bean...user can create the custom event set that handles these situations." Bean object is separate from custom event set that is created and called by bean object. Col. 8, line 62- col. 9, line 6, "The component palette displays components available in the JBuilder component library. Components are the elements which a user employs to build his or her applications...The palette can incorporate user-created custom controls....Additionally, the user can install third-party components." The bean object upon an event occurrence can message customized code sequences, separately retrievable.)

Per claims 2:

The customizable application object of claim 1, wherein said at least one of the customized code sequences comprises machine-executable binary code.

(Timbol, col. 6, lines 57-60, "...the client executes a "compiled"...program (a collection of objects) which has been created by compiling...source code (customized code sequences) ..." Also, col. 1, lines 43-45, "JAVA programs are 'compiled' into a binary format that can be executed ...")

Per claims 3:

The customizable application object of claim 1, wherein said event handler invokes said a particular one of the customized code sequences based on which of said one or more events is signified by said internal logic.

(Timbol, col. 16, lines 23-25, "The user can also make the bean a listener for events that occur in other components...", col. 16, lines 59-67, "If the bean is registered with another component as a listener...the source component calls one of the methods (Event handlers created to respond to internal logic by invoking a customized code sequence. There obviously can be a plurality of code sequences developed to handle many various events.) in the listening component when that type of event occurs...if the user wants the component to respond in some way to such an event,

the user simply writes the code that respond within the body of the KeyPressed() method...", col. 21, line 12, "...user event occurring in the IDE triggers the process.")

Per claim 4: (Timbol: Col. 15, line 58, "...adds the following three fire <event> methods...")

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 5, 6, 8-10, and 13-16** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,237,135 to Timbol, in view of U.S. Patent 6,424,979 to Livingston et al.

Timbol disclosed a component based, application development system to included customizable objects. Timbol failed to disclose supporting information concerning the database and queries thereon. However, Livingston disclosed a remote server database for an enterprise. User queries are processed from customized components of the database through event handling.

Per claim 5:

-logic which retrieves said at least one of the customized code sequences from a database which stores a plurality of customized code sequences, the retrieval being based on a query.

(Livingston: Abstract, lines 7-11, "The interface provides the user's selection of desired information within the portal in the form of a page request that is converted into queries of a database that seek content satisfying the type, level of detail and time frame attributes of the request." Also, col. 4, lines 23-25, "Each component is represented with varying levels of detail and multiple time frames allowing the view of the architecture to be customized based on user preferences.")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Timbol's component based application development system to have included information on database storage / queries as disclosed by Livingston, because Timbol suggested (col. 3, lines 37-42 & col. 3, line 63-col. 4, line 2) the need for improved code generation that allow for later, further automating of code by using components that are typically stored, accessed through databases, and manipulated in response to events (Timbol: col. 16, lines 23-67), in enterprise (Timbol: col. 20, line 7) systems. Livingston disclosed (col. 1, lines 53-61) an architectural presentation and management system that is dynamic with the ability to stay current.

Per claim 6: (Livingston: Col. 11, lines 50-67, "...generator compares the user's request (time dimension, level of detail, etc.) to the attributes stored in the XML tags that mark the tree's components, and only returns the information contained within tags whose attributes match the desired dimensions. The generator accomplishes this by using one or more queries...XML page generator sends out queries...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Timbol's component based application development system to have included information on database storage / queries as disclosed by Livingston, because components are typically stored, accessed through databases, and manipulated in response to events in a development system that is dynamic.

Per claim 8: (Livingston: Col. 7, lines 52-55, "...the relational database also allows the EAM to store custom viewing preferences and configurations for each user of the architecture..." Also, col. 10, lines 55-60, "...user positions his...mouse...on an active link...and clicks...This action invokes ...hyperlink behavior...loads a new page...or activates a procedure through a script command assigned to the event handler...The system then retrieves...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Timbol's component based application development system to included information on database storage / queries as disclosed by Livingston, because components are typically stored, accessed through databases, and manipulated in response to events in a development system that is dynamic.

Per claim 9: (Livingston: Col. 6, lines 25-30, "Applications processing the XML data can then present a subset of those units by matching the attributes with the need expressed by the user...XML enables the logical assembly of the more detailed set of information." Also col. 12, lines 58-62, "Preferably, every section in the EAM has an owner and an expiration date. When a

section reaches the expiration date, a notification agent initiates (executable action) the workflow by sending an e-mail reminder of the section owner.")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Timbol's component based application development system to included information on database storage / queries as disclosed by Livingston, because components are typically stored, accessed through databases, and manipulated in response to events in a development system that is dynamic.

Per claim 10: (Livingston: Col. 11, line 67-col. 12, line 13, "XML page generator sends out queries requesting information...XML...queries the content database...The XML page generator receives the information it previously requested")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Timbol's component based application development system to included information on database storage / queries as disclosed by Livingston, because components are typically stored, accessed through databases, and manipulated in response to events in a development system that is dynamic.

Per claim 13: (Timbol: Col. 7, lines 5-9, "...loader will unpack different sections of a file and instantiate in-memory corresponding data structures. The class loader will invoke itself recursively for loading any superclasses of the current class which is being unpacked.")

Per claim 14: (Timbol, col. 6, lines 57-60, "...the client executes a "compiled"...program which has been created by compiling...source code...")

Per claim 15: (Timbol: Col. 19, lines 16-17, "An Enterprise Java Bean is a non-visual bean that runs on a server.")

Per claims 16: (Timbol: Col. 6, lines 24-25, "Software system which is stored in system memory and / or on disk storage...")

7. **Claims 7, 11, 12, 17-49** are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,237,135 to Timbol, and in view of US Patent 6,424,979 to Livingston et al., and further in view of US Patent 6,654,029 to Chiu.

Regarding claims 7, 11, 12, 18, 19, 20, 31, 32, and 44:

Timbol disclosed a component based, application development system to included customizable objects. Livingston disclosed databases used in managing enterprise architectures. The combination failed to disclose environmental considerations and the naming conventions used to access various locations (deriving monikers from external attributes).

However, Chiu disclosed storing code in a database, retrieving code based on a query and executing code that satisfies the query. Chiu disclosed the use of a query that is based on information derived from the environment in which software operates. At col. 10, lines 41-46, "...some tools may have browsing or searching capabilities...different program resources of the

present invention will be utilized and thus different support code (for different environments) will be required." At col. 13, lines 40-44, "Querying involves describing the assets of interest...is formed by specifying the desired values for asset attributes (moniker string related to environment), including keywords. When the query is executed, a set of matching assets (objects) is returned. Chiu disclosed, (col. 4, lines 9-13, "Assets are implemented using a technique referred to herein as 'data modeling'. Data modeling provides the support for the inquiry (moniker) of assets during runtime. For example, at runtime, a multimedia production tool can determine information pertaining to any asset in the Vault." Also, col. 16, lines 38-45, "Cataloging. Different attributes...may be associated with an asset when stored in the Vault repository. Many tools derive and store much of this associated information automatically (to form moniker)..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have modified the Timbol / Livingston combined invention for customized components to include information on naming conventions (monikers) and environmental considerations used for identifying customized modules because associating attributes with an object when stored is well known in the art and supplying meaningful names for objects or modules is a logical manner for storing and accessing data through queries. All references relate to custom object code being stored and retrieved. (Chiu: Col. 1, lines 62-65, "In a typical shared development process, such inconsistencies make it difficult to coordinate and share resources among various members of the development team....such inconsistencies make it difficult to catalog and archive data in an efficient, safe, and consistent manner."

Per claims 17, 27, 36 and 38:

Timbol disclosed a component based, application development system to included customizable objects. The Timbol/Livingston combination disclosed supporting information concerning the database and queries thereon.

However, Chiu disclosed storing code in a database, retrieving code based on a query and executing code that satisfies the query. Chiu disclosed predetermined actions and at least one externally-definable action retrieved from a database query. See Chiu, col. 2, lines 34-37, "The present invention provides an integrated platform for a variety of diverse computerized utilities and application programs that operate on and/or create various types of multimedia data." And lines 63-64, "Also provided is a means for modifying and/or adding extensions (further customize or derive from base object) to the plurality of utilities and services provided..." At col. 3, lines 1-7, "...provides services and utilities for indexing, storing, retrieving , searching, and generally managing and manipulating all of the multimedia data created or used...through the use of...data base management..." At lines 11-15, "services and utilities provided...can easily be modified and/or extended. Thus, a production studio simply attaches, in a "plug-and-play'(PNP) fashion (externally-definable action), one or more supported DBMS..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified the Timbol/Livingston combination of customizable objects to include the database functionality as disclosed by Chiu, because the 'plug and play' technique allows for an integrated efficient production environment for development. It is well known in the art. Database queries, retrieval and actions are well known in the art.

Regarding claim 21,Timbol disclosed at Col. 21, lines 14-18, "This includes information about the collection of files, class path, class loader location, and the like, for the current project under development. This allows the wizard to determine…where a reference that occurs in a file may be located." The Timbol/Livingston combination failed to mention "pointers", however, Chiu disclosed the use of pointers to locate code at col. 18, lines 9-12, "…assets contents may be stored outside of the Vault repository and a pointer (e.g., file name) to where the contents are located stored as an attribute…"

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have included information regarding locating code with pointers as this is well known it the art, and thus inherent.

Per claims 22 – 24, and 28: (Timbol: Col. 4, lines 20-22, "…specify the name of the bean, the package it will be in, and the class (base object) it extends from." Col. 4, lines 25-26, "…choose a class to extend (from a base class)…" Col. 10, lines 17-20, "…once the user has created a "JAVA Bean"…can continue to use the BeansExpress…to make further changes to the generated component…" (further changes to the base object) Also, col. 9, lines 64-67, "The present invention provides a wizard-based tool which automatically generates code to define property setters and getters, accessor methods, event listener and registration mechanisms and the like."

Also, col. 16, lines 59-67, "If the bean is registered with another component as a listener…the source component calls one of the methods in the listening component when that type of event occurs…Therefore, if the user wants the component to respond in some way (public object

model which includes identifiable references to said one or more events) to such an event, the user simply writes (customize) the code that responds within the body of the ...method..." (receive the signified events...and invoke at least one customized code sequence...))

Per claims 22, 27-29, and 30: (Timbol, col. 16, lines 23-25, "The user can also make the bean a listener for events that occur in other components...", col. 16, lines 64-67, "...if the user wants the component to respond in some way to such an event, the user simply writes the code that respond within the body of the KeyPressed() method...", col. 21, line 12, "...user event occurring in the IDE triggers the process.")

Per claim 25: (Timbol, col. 6, lines 57-60, "...the client executes a "compiled"...program which has been created by compiling...source code...")

Per claim 26: (Timbol.: Col. 6, lines 24-25.)

Per claims 33, 40, 41 and 42: (Livingston, Col. 18, lines 10-55, "The first layer has a naming convention of...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified the invention of Timbol, to include database and query information as provided by Livingston and Chiu, including information on the naming (moniker) conventions used for identifying customized modules because associating attributes with an object when stored is well known in the art and supplying meaningful names for objects or

modules is a logical manner for storing and accessing data through queries. All references relate to custom object code being stored and retrieved.

Per claim 34: (Livingston: Col. 11, lines 50-67, "...generator compares the user's request (time dimension, level of detail, etc.) to the attributes stored in the XML tags that mark the tree's components, and only returns the information contained within tags whose attributes match the desired dimensions. The generator accomplishes this by using one or more queries...XML page generator sends out queries...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified the invention of Timbol, to include database and query information as provided by Livingston because associating attributes with an object when stored is well known in the art and supplying meaningful names for objects or modules is a logical manner for storing and accessing data through queries. All references relate to custom object code being stored and retrieved.

Per claims 35: (Timbol: Col. 19, lines 16-17, "An Enterprise Java Bean is a non-visual bean that runs on a server.")

Per claim 37: (Timbol, col. 6, lines 57-60, "...the client executes a "compiled"...program which has been created by compiling...source code...")

Per claims 40, 41 and 42: (Livingston: Fig. 1 and col. 9, lines 39-49, "...The request is

provided to a web server which requests the needed information from an object server...")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified the invention of Timbol, to include database and query information as provided by Livingston, including requests handled by a web server because associating attributes with an object when stored is well known in the art and client / server arrangements for storing / accessing modules is a logically done through queries. This is well known in the art.

8. **Claim 50** is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,237,135 to Timbol.

Regarding claim 50, Timbol disclosed:

A system for performing a customizable task comprising:

(Timbol: col. 4, lines 6-7, "The present invention provides a development system with...visual designers...")

-means for performing one or more predetermined actions;

(Timbol: Col. 4, lines 41-42, "...using the visual designers to manage the bean's properties ...", col. 9, lines 36-37, "...standard design patterns for declaring properties, methods (predetermined actions)...")

-means for signifying one or more events;

(Timbol: Col. 15, lines 12-14, "A JAVA Bean can generate (or fire) events, sending an event object to a listening object, and can listen for events and respond to the m when they occur.")

-means for storing a plurality of code modules;

(Timbol: Col. 7, lines 25-29, "Runtime support libraries (stored code modules) comprise functions which provide runtime support to the virtual machine, including memory management, synchronization, type checking, and interface invocation", col. 8, lines 61- col. 9, line 6, "The component palette displays components available in the JBuilder component library...Components are the elements which a user employs to build his or her application...the user can install third-party components." The means exist for storing code modules.)

-means for loading a selected one of said plurality of code modules;

(Timbol: Col. 8, lines 30-32, "The bytecode which is executed repeatedly calls (loads a code module) into the runtime support libraries, for invoking various JAVA runtime functions."

-means for invoking at least a portion of the selected code module in response to said one or more events, the code module being selected based, on one or more factors comprising aspects of an environment in which said code module executes, said aspects being external to an object that causes said code module to be invoked.

(Timbol: Col. 15, lines 26-28, "These methods are called (invoke selected code) by components that want to be notified when these types of events occur." In reference to the selection of the code module, the designer registers a code module to a component. Col. 15, lines 23-24, "user selects the events he or she wants the bean capable of firing..." The event handler code is external to the component. Timbol disclosed that environmental issues could be considered. . Col. 5, lines 62-66, "The present invention, however, is not limited to any particular application or any particular environment...may be advantageously applied to a variety of platforms and

environments..." Col. 6, lines 44, "may be implemented in other platforms...", col. 8, line 9, "Displaying the Environment Options dialog")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to have modified Timbol's invention for customized components to include the selection of code modules based on environmental aspects, as Timbol had provided for a variety of environments. Motivation may be found at col. 1, lines 31-36, "With the explosive growth of the Internet and the World Wide Web, an ever-increasing number of computers of disparate platforms are being connected together. As a result there is renewed interest in distributing software in binary format which operates in the ever-increasing heterogeneous environment."

9. ***Response to Arguments***

(A) Applicant has argued, in substance, the following:

Regarding independent Claim 1, as Applicant has noted on page 14, 2nd paragraph, of Amendment, dated 26 April 2004,Timbol does not teach the limitations of claim 1: "invokes at least one of a plurality of customized code sequences based on said data or logic, each of the at least one customized code sequence being separately retrievable from a source separate from said customization object".

Examiner' Response:

This is a newly added limitation and is addressed in the claim 1 rejection above.

(B) Applicant has argued, in substance, the following:

Regarding independent claim 9, as Applicant has noted in the last paragraph of page 14 and the 1st paragraph of page 15, of Amendment dated 26 April 2004, "Livingston does not teach the selection of a custom code module that comprises executable components." Timbol "does not teach that code module (or bean) is selected from a plurality of code modules that contain code to perform variable actions." Thus the combination fails to teach the limitations of claim 9.

Examiner' Response:

Timbol discloses that the user / designer relates event handler code to a specific component, (col. 15, lines 23-24) "user selects the events he or she wants the bean capable of firing...",(col. 4, line 42) "by using the visual designers to manage the bean's properties....". Timbol disclosed, col. 16, lines 64-67, "if the user wants the component to respond in some way to such an event, the user simply writes the code that responds within the body of the KeyPressed() method, for example." Thus Timbol selects a custom code module...comprising on or more executable components.

Livingston disclosed an invention to provide users customized views by retrieving desired components. Col. 2, lines 6-7, "relevant information components...based on the user's request..." Components (custom code module) are selected through an XML query framework. Col. 6, lines 22-30, "The EAM uses XML to assemble the units of information about the technical architecture into meaningful documents... "Applications processing the XML data can then present a subset of those units by matching the attributes with the need expressed by the user...XML enables the logical assembly of the more detailed set of information." Also col. 12,

lines 58-62, "Preferably, every section in the EAM has an owner and an expiration date. When a section reaches the expiration date, a notification agent initiates (executable action) the workflow by sending an e-mail reminder of the section owner." Thus, Livingston does teach the selection of a custom code module that comprises executable components.

(C) Applicant has argued, in substance, the following:

Regarding independent claim 17, as Applicant has noted in the last paragraph of page 15, the combined references of Timbol, Livingston and Chiu fails to teach or suggest "the retrieval of a code module from a database."

Examiner' Response:

Timbol disclosed the retrieval of a code module. Col. 7, lines 5-9, ...loader will unpack different sections of a file and instantiate in-memory corresponding data structures. The class loader will invoke itself recursively for loading any superclasses of the current class which is being unpacked." Timbol disclosed that the (col. 6, lines 48-51) "Application software can be any one of a variety of software applications, such as...database..." Code modules are retrieved for loading.

(D) Applicant has argued, in substance, the following:

Regarding independent claim 44, as Applicant has noted in the last paragraph of page 17, the combination of Timbol, Livingston, and Chiu, "do not teach or suggest the storage or retrieval of code in a database" nor "teach the storage or retrieval of a pointer to code in a database." Additionally, as noted in the 2nd paragraph of page 18, the combined references fail to disclose

the limitation of "...'ascertaining one or more attributes of said operating environment external to said software object,' where the query to the database is based on the attributes."

Examiner' Response: Timbol, in view of Livingston is used to reject claim 44. Timbol disclosed a component based application development system, including customizable objects. Timbol suggested a database, but failed to include information concerning queries thereon. Timbol suggested allowing for various environments. See Col. 8, line 9, Tools menu: Commands for : Displaying the Environment Options dialog" Timbol disclosed that components are registered to listeners and related event handlers (pointers to event handler code) are called.

Livingston disclosed a remote server database of an enterprise. User queries are processed from customized components of the database through event handling. Col. 23, lines 14-21, "The present invention has been described with respect to a system that is optimized for presenting and managing dynamic information. The invention can be extended to any other set of information...or applied to any sources of information that can be divided and managed as atomic units and assembled and presented in varying constructs and contexts."

(E) Applicant has argued, in substance, the following:

Regarding independent claim 50, as Applicant has noted in the 4th paragraph of page 18, the combined references do not teach or suggest the use of environmental aspects to select a code module.

Examiner' Response: Timbol does environmental features. See the rejection of claim 50 above.

(F) Applicant has argued, in substance, the following:

Regarding the dependent claims, as Applicant has noted in the 2nd, 3rd, and 4th paragraphs of page 19, the combined references fail to disclose:

-the retrieval of code from a database;

-for the information to be derived from the environment and used as part of a database query;

-information to be used to construct moniker string that is used as part of the query;

-the retrieval, from a database of a pointer to code.

Examiner' Response:

-Timbol disclosed the invention could include a database. Event handler code is retrieved as required. Livingston provided for storing and retrieving code modules in a database. Chiu disclosed a database (vault) and querying to retrieve assets.

- Event handler code is retrieved as required. Timbol disclosed Environmental Options.

- Chiu disclosed (col. 16, lines 38-45, "Different attributes, e.g., type, date created, and keywords, may be associated with an asset (information to be used to construct moniker string) when stored in the Vault repository. Many tools derive and store much of this associated

information automatically on behalf of the user. Examples of such derived information may include the date and time an asset is created, as well as the identity of the creator.

- Timbol disclosed that components are registered to listeners and related event handlers (pointers to event handler code) are called. Livingston disclosed a remote server database of an enterprise. User queries are processed from customized components of the database through event handling. Col. 23, lines 14-21, "The present invention has been described with respect to a system that is optimized for presenting and managing dynamic information. The invention can be extended to any other set of information...or applied to any sources of information that can be divided and managed as atomic units and assembled and presented in varying constructs and contexts."

Examiner maintains the rejections of claims 1-50. Applicant has claimed basic JAVA component messaging, storing, and retrieval. It is well known that the components may be customized. It is well known that event handlers can be customized. Storing and retrieving components from a database is well known.

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (703) 305-4564. The examiner can normally be reached Monday through Thursday, from 7:00 A.M. to 5:30 P.M. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (703) 305-4552.

The fax phone number is (703) 872-9306 for regular communications and for After Final communications. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Mary Steelman

07/19/2004

WEI Y. ZHEN
PRIMARY EXAMINER